**《面向对象程序设计语言》作业（3.26）**

地信班109092023XXX 许愿

1. 找出一个二维数组中的鞍点，即该位置上的元素在该行上最大，在该列上最小（也可能没有鞍点）。

#include <iostream>

using namespace std;

int main(){

    const int n = 5, m = 5; // 行数和列数

    int a[n][m] = {{3,2,4,8,9},{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15},{1,5,7,3,5}}; // 定义数组

    int i, j, k, l; // 列号存储

    int max, min;

    for(i = 0; i < n; i++){ // 行内

        max = a[i][0]; // 初始化最大值为第一列的值

        k = 0;

        for(j = 1; j < m; j++){ // 列内

            if(a[i][j] > max){ // 若有比当前最大值大的值

                max = a[i][j]; // 更新最大值

                k = j;

            }

        }

        // 最终得到本列最大值为max, 列号为k

        min = a[0][k];

        l = 0;

        for(j = 1; j < n; j++){

            if(a[j][k] < min){

                min = a[j][k];

                l = j;

            }

        }

        // 最终得到本行最小值为min, 行号为l

        // 判断是否为鞍点

        if(l == i){ // 同时满足最大和最小

            cout << "鞍点为：" << a[i][k] << "，位置为：" << i << "," << k << endl;

            return 0;

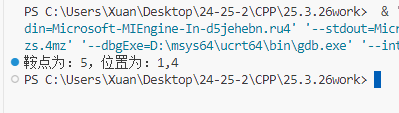
        }

    }

    cout << "没有鞍点" << endl; // 没有鞍点

    return 0;

}



1. 有15个数按从大到小顺序存放在一个数组中，输入一个数，要求用折半查找法找出该数是数组中第几个元素的值。如果该数不在数组中，则打印出“无此数”。

#include <iostream>

using namespace std;

int main(){

    int a[15] = {99, 88, 77, 66, 55, 44, 33, 22, 11, 10, 9, 8, 7, 6, 5}; // 定义数组

    int num; // 输入的数

    cout << "请输入一个数：";

    cin >> num;

    int left = 0;

    int right = 14; // 15-1

    while(left <= right){ // 左边界小于等于右边界

        int mid = (left + right) / 2; // 取出中间位置

        if(a[mid] == num){ // 若找到数

            cout << "该数在数组中的位置为：" << mid + 1 << endl; // 输出数在数组中的位置

            return 0;

        } else if(a[mid] < num){ // 中间值小于输入的数，左半部分查找

            right = mid - 1;

        } else { // 中间值小于输入的数，右半部分查找

            left = mid + 1;

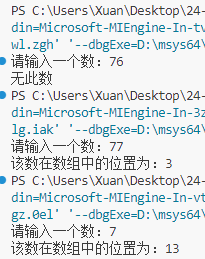
        }

    }

    cout << "无此数" << endl; // 没有找到

    return 0;

}



1. 给出年、月、日，计算该日是该年的第几天。

#include <iostream>

using namespace std;

int main(){

    int year, month, day; // 年、月、日

    cout << "请输入年、月、日：";

    cin >> year >> month >> day;

    int days = 0; // 初始化天数

    for(int i = 1; i < month; i++){ // 遍历月份

        switch(i){ // 判断月份

            case 1: case 3: case 5: case 7: case 8: case 10: case 12: // 大月

                days += 31;

                break;

            case 4: case 6: case 9: case 11: // 小月

                days += 30;

                break;

            case 2: // 2月特殊判断

                if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)){ // 闰年的2月为29天

                    days += 29;

                } else { // 否则为2月28天

                    days += 28;

                }

                break;

        }

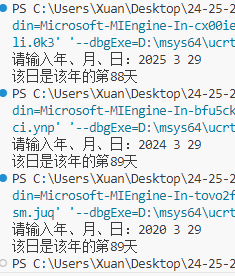
    }

    days += day; // 加上当前月已过的天数

    cout << "该日是该年的第" << days << "天" << endl; // 输出结果

    return 0;

}



1. 有一篇文章，共有3行文字，每行有80个字符。要求分别统计出其中英文大写字母、小写字母、数字、空格及其他字符的个数。

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    string data = "BdOFkx\*)KicbHD#XS$cCUML$B~B+cXe  BZ\*NO}%FArptY/U/ujiWQdaK%t &XwwHZ?w\_Z=[=/%YFQ,V\n<RO)LnK/|fNblSjyezxDN.cze{G+l<qET,G@X%?CwS&Eoh@N/MbiurTiab:!+eNk-,{W[pibZ!Ldt+>E\n--FrZkHPrTJ^I~v?AdmFVg@B/Wls~<;yl@T/{W/Ia(v#|UoVUnApxU:mhh}/&GkTj^VFYo|/)a?wKK^v";

    int cnt\_dxzm, cnt\_xxzm, cnt\_sz, cnt\_gk, cnt\_qtzf = 0;

    for(int i = 0; i < data.length(); i++){

        if(data[i] >= 'A' && data[i] <= 'Z'){ // 大写字母

            cnt\_dxzm++;

        } else if(data[i] >= 'a' && data[i] <= 'z'){ // 小写字母

            cnt\_xxzm++;

        } else if(data[i] >= '0' && data[i] <= '9'){ // 数字

            cnt\_sz++;

        } else if(data[i] == ' '){ // 空格

            cnt\_gk++;

        } else { // 其他字符

            cnt\_qtzf++;

        }

    }

    cout << "大写字母个数：" << cnt\_dxzm << endl;

    cout << "小写字母个数：" << cnt\_xxzm << endl;

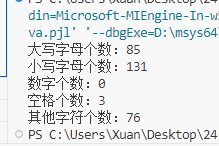
    cout << "数字个数：" << cnt\_sz << endl;

    cout << "空格个数：" << cnt\_gk << endl;

    cout << "其他字符个数：" << cnt\_qtzf << endl;

    return 0;

}



1. 有一行电文，已按下面规律译成密码：A-Z, a-z, B-Y, b-y, C-X, c-x, .. 即第1个字母变成第26个字母，第i个字母变成第(26-i+1)个字母...非字母字符不变。要求编程序将密码译回原文，并打印出密码和原文。

#include <iostream>

using namespace std;

int main(){

    string dwmm = "wfdaujfvgaYWGDADydgawesuydhQQSDYAwyhuixdahuicha283164r2fcgvy8xd8swacy7";

    string dw = "";

    for(int i = 0; i < dwmm.length(); i++){

        if(dwmm[i] >= 'A' && dwmm[i] <= 'Z'){ // 大写字母

            dw += 'A' + ('Z' - dwmm[i]);

        } else if(dwmm[i] >= 'a' && dwmm[i] <= 'z'){ // 小写字母

            dw += 'a' + ('z' - dwmm[i]);

        } else { // 非字母字符

            dw += dwmm[i];

        }

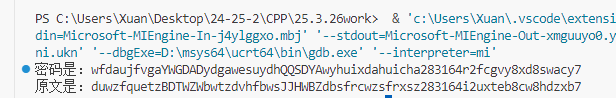
    }

    cout << "密码是：" << dwmm << endl;

    cout << "原文是：" << dw << endl;

    return 0;

}



1. 输入n个字符串，将它们按字母由小到大的顺序排列并输出。

#include <iostream>

using namespace std;

int main(){

    int n;

    cout << "请输入要输入的字符串数量: ";

    cin >> n;

    string\* str = new string[n];

    for(int i = 0; i < n; i++){

        cout << "请输入第" << i + 1 << "个字符串: ";

        cin >> str[i];

    }

    // 排序

    for(int i = 0; i < n - 1; i++){

        for(int j = 0; j < n - i - 1; j++){

            if(str[j] > str[j + 1]){

                string temp = str[j];

                str[j] = str[j + 1];

                str[j + 1] = temp;

            }

        }

    }

    // 输出

    cout << "排序后的字符串为: " << endl;

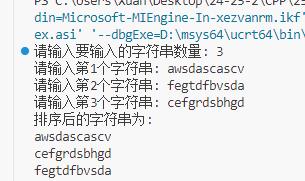
    for(int i = 0; i < n; i++){

        cout << str[i] << endl;

    }

    return 0;

}



1. 输入n个字符串，把其中以字母A打头的字符串输出。

#include <iostream>

using namespace std;

int main(){

    int n = 0;

    cout << "请输入字符串个数: ";

    cin >> n;

    string\* str = new string[n]; // 生成字符串数组

    for(int i = 0; i < n; i++){

        cout << "请输入第" << i + 1 << "个字符串: ";

        cin >> str[i]; // 输入字符串

    }

    cout << "A开头的字符串有: " << endl;

    for(int i = 0; i < n; i++){

        if(str[i][0] == 'A' || str[i][0] == 'a'){ // 首字母开头为A或a

            cout << str[i] << endl;

        }

    }

    return 0;

}



17. 输入10个学生的姓名、学号和成绩，将其中不及格者的姓名、学号和成绩输出。

#include <iostream>

using namespace std;

int main(){

    const int n = 10; // 总共10名学生

    string\* name = new string[n]; // 姓名

    string\* id = new string[n]; // 学号

    int\* score = new int[n]; // 成绩

    for(int i = 0; i < n; i++){

        cout << "请输入第" << i + 1 << "个学生的姓名: ";

        cin >> name[i]; // 输入姓名

        cout << "请输入第" << i + 1 << "个学生的学号: ";

        cin >> id[i]; // 输入学号

        cout << "请输入第" << i + 1 << "个学生的成绩: ";

        cin >> score[i]; // 输入成绩

    }

    cout << "不及格的学生有: " << endl;

    for(int i = 0; i < n; i++){

        if(score[i] < 60){ // 成绩不及格

            cout << "姓名: " << name[i] << ", 学号: " << id[i] << ", 成绩: " << score[i] << endl;

        }

    }

    return 0;

}

